Claims:

This listing of claims will replace prior versions, and listings of claims in the application.

1. (Previously Presented) A method for accessing status information related to a process the method comprising:

receiving a request from a client for status information related to the process;

identifying nodes in a network, each of the nodes executing a distributed thread of the process;

polling each identified node for status information associated with the thread executing by the node, the status information generated by a script associated with the process;

receiving the status information from each of the nodes;

storing the status information in a data structure; and

enabling the client to access the status information.

3. (Previously Presented) The method of claim 1, further comprising:

invoking one or more script engines to execute at least one script code that performs at least one action of the process;

handling multiple script threads during the execution of the process.

4. (Previously Presented) The method of claim 3, wherein the one or more script engines are maintained by a process management system that executes on the nodes.

5. (Previously Presented) The method of claim 4, wherein the one or more nodes include a primary node.

6. (Previously Presented) The method of claim 1, further comprising making the data structure available to any node in the network capable of accessing a process management system in a primary node.

7. (Previously Presented) The method of claim 6, wherein the step of polling is performed by the process management system residing on the primary node over an established connection with the identified nodes.

8. (Previously Presented) The method of claim 7, wherein the identified nodes include the primary node.

12. (Previously Presented) The method of claim 1, wherein the step of storing is performed by a process management system executing on a primary node.

13. (Previously Presented) The method of claim 12, wherein the step of storing further includes:

placing the status information relative to the executable process into a private data structure by the process management system on the primary node, wherein the private data structure is accessible to only script threads that are spawned during the execution of the process.

14. (Previously Presented) The method of claim 12, wherein the step of storing further includes:

placing the status information relative to the executable process into a status value data structure that is accessible to any node capable of accessing the process management system executing on the primary node.

15. (Previously Presented) The system of claim 14, wherein the status value data structure comprises data for providing an indication of an event that occurs during the execution of the process.

16. (Previously Presented) The method of claim 1, further comprising:

establishing a connection between a process management system executing on at least one of the nodes and another process management system residing on a

primary node, wherein the connection is established by a script code in execution by the a script engine associated with the at least one node.

17. (Previously Presented) The method of claim 1, further comprising:

establishing a connection between other client nodes and a process management system residing on a primary node, wherein the connection is established from a user interface executing on the other client nodes; and

accessing the process management system from over the established connection by the user interface executing on the other client nodes.

18. (Original) The method of claim 17, wherein the step of establishing includes accepting a command as input by the user interface to establish a connection with the process management system executing on the primary node.

19. (Original) The method of claim 17, wherein the step of accessing includes accepting a command as input by the user interface to invoke the action of the executable process by the process management system from over the established connection.

20. (Original) The method of claim 17, wherein the step of accessing includes accepting a command as input by the user interface to poll the process

management system for status information from over the established connection.

21. (Original) The method of claim 17, wherein the user interface receives messages from the process management system over the established connection.

22. (Original) The method of claim 21, wherein the messages contain information that is descriptive of the primary node.

23. (Original) The method of claim 21, wherein the messages contain information that is descriptive of a particular event that occurs during the execution of the process.

24. (Original) The method of claim 21, wherein the messages contain a data structure that is generated as a result of the execution of the script code by the one or more script engines to indicate the status of the executable process.

42. (Previously Presented) A system comprising:

a process management system executing on a primary node in a network, the process management system configured to collect status information associated with a process, the processing management system also configured

to divide the process into multiple threads and distribute the threads to multiple remote nodes in the network, the process management system further configured to receive the status information associated with the threads from each remote node and store the status information in a data structure accessible by any node with authorized access to the process management system; and

the remote nodes in the network, each remote node processing at least one of the threads associated with the process and including a script configured to provide the status information collected by the process management system.

43. (Previously Presented) The system of claim 42, further comprising one or more client node each configured with a user-interface, the one or more user interfaces configured to establish a connection over the network with the process management system executing on the primary node, the one or more user interfaces also configured to request the status information from the process management system and to process the status information when the information is received.

46. (Original) The system of claim 42, wherein the one or more user interfaces accept as input commands to establish a connection with the process management system executing on the primary node.

47. (Original) The method of claim 42, wherein the one or more user interfaces accept as input commands to invoke the action of the executable process by the process management system, and sends requests to invoke the action of the executable process to the process management system from over the established connection.

48. (Previously Presented) The system of claim 42, wherein the one or more user interfaces accept as input commands to poll the process management system for status information, and sends requests to poll the process management system for status information from over the established connection.

49. (Original) The system of claim 42, wherein the one or more user interfaces receive messages from the process management system over the established connection in response to the polling.

50. (Previously Presented) The system of claim 49, wherein the messages contain information that is descriptive of the primary node.

51. (Previously Presented) The system of claim 49, wherein the messages contain information that is descriptive of a particular event that occurs during the execution of the process.

52. (Previously Presented) The system of claim 49, wherein the messages contain a data structure that is generated as a result of the execution of the script code by the one or more script engines to indicate the status of the executable process.

53. (Previously Presented) The system of claim 42, wherein the process management system accepts connection requests from one or more user interfaces operating on one or more nodes associated with the process management system over an established connection.

54. (Original) The system of claim 53, wherein the one or more nodes include the primary node.

55. (Original) The system of claim 42, wherein the process management system receives requests to invoke the action of the executable process from the one or more nodes connected to the process management system.

56. (Original) The system of claim 42, wherein the process management system continuously polls the one or more nodes connected to the process management system to obtain status information related to the executable process.

57. (Previously Presented) The system of claim 42, wherein the process management system stores the information into a public data structure that is accessible to the one or more nodes capable of establishing a connection with the process management system.

58. (Previously Presented) The system of claim 42, wherein the process management system stores the status information relative to the process into a private data structure that is accessible to only script threads in operation during process execution.

59. (Original) The system of claim 42, wherein the process management system stores the status information relative to the executable process into a status value data structure that is accessible to the one or more nodes having access to the status information.

60. (Original) The system of claim 59, wherein the status value data structure contains data for providing an indication of a particular event that occurs during the execution of the process.

61. (Original) The system of claim 42, wherein the process management system receives requests for status information relative to the executable process from the one or more nodes connected to the process management system.

62. (Original) The system of claim 42, wherein the process management system sends the public data structure to the one or more nodes in response to the request.

63. (Original) The system of claim 42, wherein the process management system sends the status value data structure to the one or more nodes in response to the request.

64. (Original) An apparatus comprising:

means for receiving a request from a client to initiate a process;

means for dividing the process into multiple threads;

means for distributing the threads to multiple nodes in a network for execution;

means for polling each node for status information generated by a script executing in the node;

means for receiving the status information from each of the nodes;

means for storing the status information in a data structure; and

means for enabling any node with authorization to access the status information.